

---

GemStone®

***GemBuilder® for Java™***  
***Release Notes***

Version 3.0

September 2010

GEMSTONE ™

---

## **INTELLECTUAL PROPERTY OWNERSHIP**

This documentation is furnished for informational use only and is subject to change without notice. GemStone Systems, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this documentation.

This documentation, or any part of it, may not be reproduced, displayed, photocopied, transmitted, or otherwise copied in any form or by any means now known or later developed, such as electronic, optical, or mechanical means, without express written authorization from GemStone Systems, Inc.

Warning: This computer program and its documentation are protected by copyright law and international treaties. Any unauthorized copying or distribution of this program, its documentation, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted under the maximum extent possible under the law.

The software installed in accordance with this documentation is copyrighted and licensed by GemStone Systems, Inc. under separate license agreement. This software may only be used pursuant to the terms and conditions of such license agreement. Any other use may be a violation of law.

Use, duplication, or disclosure by the Government is subject to restrictions set forth in the Commercial Software - Restricted Rights clause at 52.227-19 of the Federal Acquisitions Regulations (48 CFR 52.227-19) except that the government agency shall not have the right to disclose this software to support service contractors or their subcontractors without the prior written consent of GemStone Systems, Inc.

This software is provided by GemStone Systems, Inc. and contributors "as is" and any expressed or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall GemStone Systems, Inc. or any contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

## **COPYRIGHTS**

This software product, its documentation, and its user interface © 1986-2010 GemStone Systems, Inc. All rights reserved by GemStone Systems, Inc.

## **PATENTS**

GemStone is covered by U.S. Patent Number 6,256,637 "Transactional virtual machine architecture", Patent Number 6,360,219 "Object queues with concurrent updating", and Patent Number 6,567,905 "Generational Garbage Collector". GemStone may also be covered by one or more pending United States patent applications.

## **TRADEMARKS**

**GemStone**, **GemBuilder**, **GemConnect**, and the GemStone logos are trademarks or registered trademarks of GemStone Systems, Inc. in the United States and other countries.

**UNIX** is a registered trademark of The Open Group in the United States and other countries.

**GemStone**, **GemBuilder**, **GemConnect**, and the GemStone logos are trademarks or registered trademarks of GemStone Systems, Inc. in the United States and other countries.

**UNIX** is a registered trademark of The Open Group in the United States and other countries.

**Java** and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Other company or product names mentioned herein may be trademarks or registered trademarks of their respective owners. Trademark specifications are subject to change without notice. All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized to the best of our knowledge; however, GemStone cannot attest to the accuracy of all trademark information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

**GemStone Systems, Inc.**  
1260 NW Waterhouse Avenue, Suite 200  
Beaverton, OR 97006

## About This Documentation

These release notes describe new features and bugs fixed in the GemBuilder® for Java™ Version 3.0 release. We recommend that everyone using GemBuilder for Java read these release notes before beginning installation or development.

For installation instructions, see the *GemBuilder for Java Installation Guide*.

These release notes are also available on the GemStone website, as described in the next section.

GemBuilder for Java tools may be used against any GemStone/S server product, including GemStone/S, the 32-bit server product, or GemStone/S 64 Bit. The term “GemStone” may be used to refer to any GemStone server product, as well as to the company, GemStone Systems, Inc.

## Other Useful Documents

- ▶ The *Javadocs* that are provided with the GemBuilder for Java product distribution are a key source of up-to-date information on GemBuilder for Java functionality.
- ▶ *GemBuilder for Java Tools Guide* describes the independent set of tools that let you explore and modify Smalltalk code in the server.
- ▶ *GemBuilder for Java Programming Guide* describes how to develop java based applications that interface with a GemStone server.
- ▶ The *Programming Guide* for GemStone/S or GemStone/S 64 Bit describes the GemStone System and the GemStone Smalltalk language.
- ▶ If you will be acting as a system administrator, or developing software for someone else who must play this role, you should read the *System Administration Guide* for GemStone/S or GemStone/S 64 Bit.

## Technical Support

GemStone's Technical Support website provides a variety of resources to help you use GemStone products.

**GemStone Web Site:** <http://support.gemstone.com>

Use of this site requires an account, but registration is free of charge and provides immediate access.

All GemStone product documentation is provided in PDF form on this website. Documentation is also available at

**<http://www.gemstone.com/documentation>**

In addition to documentation, the [support.gemstone.com](http://support.gemstone.com) website provides:

- ▶ Bugnotes, identifying performance issues or error conditions that you may encounter when using a GemStone product.
- ▶ TechTips, providing information and instructions that are not otherwise included in the documentation.
- ▶ Compatibility matrices, listing supported platforms for GemStone server product versions.

This material is updated regularly; we recommend checking this site on a regular basis.

## Help Requests

You may need to contact Technical Support directly, if your questions are not answered in the documentation or by other material on the Technical Support site.

Requests for technical assistance may be submitted online, or by email or by telephone. We recommend you use telephone contact only for more serious requests that require immediate evaluation, such as a production system down. The support website is the preferred way to contact Technical Support.

**Website:** <http://techsupport.gemstone.com>

**Email:** [support@gemstone.com](mailto:support@gemstone.com)

**Telephone:** (800) 243-4772 or (503) 533-3503

Your GemStone support agreement may identify specific designated contacts who are responsible for submitting all support requests to GemStone. If so, please submit your information through those individuals.

If you are reporting an emergency by telephone, select the option to transfer your call to the Technical Support administrator, who will take down your customer information and immediately contact an engineer. Non-emergency requests received by telephone will be placed in the normal support queue for evaluation and response.

When submitting a request, please include the following information:

- ▶ Your name, company name, and GemStone server license number.
- ▶ The versions of all related GemStone products, and of any other related products, such as client Smalltalk products.

- ▶ The operating system and version you are using.
- ▶ A description of the problem or request.
- ▶ Exact error message(s) received, if any, including log files if appropriate.

Technical Support is available from 8am to 5pm Pacific Time, Monday through Friday, excluding GemStone holidays.

## **24x7 Emergency Technical Support**

GemStone offers, at an additional charge, 24x7 emergency technical support. This support entitles customers to contact us 24 hours a day, 7 days a week, 365 days a year, for issues impacting a production system. For more details, contact your GemStone account manager.

## **Training and Consulting**

Consulting and training for all GemStone products are available through GemStone's Professional Services organization. GemStone periodically offers training courses at our offices in Beaverton, Oregon, or training can be arranged at your location. Contact your GemStone account representative for more details or to obtain consulting services.



***Chapter 1. GemBuilder for Java 3.0 Release Notes***

Overview of Changes . . . . . 9

- New interface layer. . . . . 9
  - GbjBroker no longer used . . . . . 10
  - Simpler design, improved transport performance . . . . . 10
- Automated Export Set Management . . . . . 10
- Object Change Notification . . . . . 10
- Explicit Object Canonicalization . . . . . 10
- Shared Libraries now required . . . . . 10
  - Can no longer simultaneously connect to multiple server products . . . . . 11
- Package com.gemstone.gbjgci . . . . . 11
  - Added GbjGci Classes . . . . . 11
- Package com.gemstone.gbj . . . . . 12
  - Logging . . . . . 12
  - Obsolete Classes . . . . . 12
  - GbjObject . . . . . 12
    - Object Change Notification . . . . . 12
    - Changes to Fields . . . . . 13
    - Changes to Methods . . . . . 13
    - Obsolete Methods. . . . . 13
    - Deprecated Methods . . . . . 14
    - Moved Methods. . . . . 14
  - GbjSession. . . . . 14
    - Changes to Fields . . . . . 14
    - Obsolete Methods. . . . . 14
    - Modified Methods . . . . . 15
    - Deprecated Methods . . . . . 15
  - GbjParameters . . . . . 15

GbjKernelObjects . . . . .	15
GbjIterator . . . . .	15
GbxLogger . . . . .	16
Deprecated Classes . . . . .	16
Package com.gemstone.tools . . . . .	16



# *GemBuilder for Java 3.0*

## *Release Notes*

GemBuilder for Java (GBJ) version 3.0 is a new release of the GemBuilder for Java product. This version is based on a new design, using the GCI interface (the interface used by GemBuilder for Smalltalk). The GbjBroker and the Gem-pooling components have been removed, resulting in a cleaner, more performant interface.

This version of GemBuilder for Java supports both 32-bit GemStone/S and GemStone/S 64 Bit.

These release notes provide details on the changes in this release. Please take time to read through them before installing the product, to acquaint yourself with the changes. There have been API changes; if you are upgrading, you will have to review your code and update as required. You will also need to review your application management, since the GbjBroker and associated initialization are no longer included.

To install GemBuilder for Java version 3.0, follow the instructions in the *GemBuilder for Java Installation Guide* for v3.0. The installation and configuration process has also changed significantly.

## **Overview of Changes**

### **New interface layer**

GemBuilder for Java version 3.0 includes a new package, GbjGci, providing an interface via java JNI code calling the GemBuilder for C Interface (GCI). This replaces the transport layer provided in previous versions, which used the GbjBroker and object marshalling.

The GbjGci package includes added classes, such as GbjGciObject and GbjGciSession, which are analogous to the GbjObject and GbjSession. Much of the core functionality has been moved to the GbjGci versions. The Gbj classes now extend the GbjGci classes.

There have been changes to the API of the Gbj classes, especially GbjObject and GbjSession. See the documentation on the specific classes in the section “Package com.gemstone.gbj” on page 12.

A number of GBJ classes originally needed to support the original object marshalling scheme are no longer needed and have been removed.

### **GbjBroker no longer used**

The GbjBroker is no longer used. The GbjBroker scripts and the associated gbj.ini file are no longer provided or needed. The broker-related fields in the GbjParameters class, and the GbjLoginDialog, have been removed.

### **Simpler design, improved transport performance**

Removing the classes and code necessary to support the gem broker mechanism and socket-based object marshalling from GBJ provides significant performance improvements and easier system management. It also allows support for other new features.

## **Automated Export Set Management**

Objects brought over to the java client are automatically added to the export set, and removed with java finalizes the object. The application no longer needs to explicitly track this.

## **Object Change Notification**

GbjObject now inherits from java class Observable, allowing the developer to define Observer objects that can be notified when the GemStone server object represented by a given client GbjObject is changed due to transactional activity or GemStone Smalltalk code execution.

## **Explicit Object Canonicalization**

Objects brought over for a particular session are explicitly canonicalized, so there always only a single GbjObject for a given server OOP. This was necessary to support a number of the new features.

## **Shared Libraries now required**

GBJ now requires access to the GemStone server GCI shared library files, and a GBJ shared library file. You will need to copy these libraries from the server distribution to a location accessible by the machine on which you are running GBJ.

Newer versions of GemStone/S and GemStone/S 64 Bit 64 include the GbjGci shared library, \$GEMSTONE/lib/libgbjgci30.so (Unix and Linux) or %GEMSTONE%\lib\gbjgci30.dll (Windows). If you are using an older version that does not include this library, contact GemStone Technical Support to request a copy. Please include your platform (Solaris, HP, AIX, Linux, or Windows), product (GemStone/S or GemStone/S 64 Bit) and product version number with your request.

## Can no longer simultaneously connect to multiple server products

In previous versions, it was possible for different GbjSessions to connect to GemStone servers with different OOP formats; for example, a GemStone/S (32-bit) session could connect at the same time as a GemStone/S 64 Bit session. The new GCI-based design constrains all GbjSessions within a java program to belong to a single product type.

## Package `com.gemstone.gbjgci`

The GbjGci package provides a new low-level interface to the GemBuilder for Java product. The previous transport layer using the gem broker and object marshalling scheme is replaced with code calling the GemBuilder for C Interface (GCI), which is also used in the GemBuilder for Smalltalk product.

The Gbj classes, such as GbjObject, now extend these new GbjGci classes.

This new layer required changes in the Gbj API. For details, see the documentation for the specific classes in the section “Package `com.gemstone.gbj`” on page 12.

The GbjGci package can also be used stand-alone, providing sufficient functionality for basic java/GS applications with a smaller code footprint and less overhead than that required by the full GBJ package.

## Added GbjGci Classes

The following classes have been added to provide the connection between the original `com.gemstone.gbj` package and the GCI C layer:

### **GbjGciInterface**

GbjGciInterface provides connectivity between java and the C-based GCI code, using the java JNI mechanism. The public API provides access to private lower level JNI methods and smooths out the flow of objects and data and exception handling.

### **GbjGciReport**

GbjGciReport represents basic object information being transferred to/from the GCI layer. To provide support for object change notification, this class extends java class `Observable`.

### **GbjGciObject**

GbjGciObject provides a slightly higher object representation that indicates which session is associated with a given object. It extends GbjGciReport.

The original GbjObject class now extends GbjGciObject, and some of its original functionality has been moved to this layer.

### **GbjGciSession**

GbjGciSession represents basic session information. It includes support for GbjGciObject canonicalization, java/GbjGciObject class mapping, export set management, and changed object updating.

The original GbjSession class now extends GbjGciSession, and some of its original functionality has been moved to this layer.

### GbjGciPreferences

GbjGciPreferences uses java Preferences to determine debug trace logging levels for the various Gbj/GbjGci modules. Customer applications should not need to change these unless requested by GemStone Technical Support to assist in debugging a Gbj/GbjGci problem.

### GbjGciException

The original GbjException class does not extend GbjGciException, but takes GbjGciExceptions and processes them as appropriate to pass on to existing GBJ code.

## Package com.gemstone.gbj

Changes in version 3.0 in the com.gemstone.gbj package include:

### Logging

Prior to version 3.0, a number of GBJ classes included the private field logger, which held an instance of com.gemstone.org.apache.log4j.logger for use in tracking diagnostic messages. This field has been made static and is replaced with java.util.logging.Logger.

### Obsolete Classes

The following classes have become obsolete and have been removed from the com.gemstone.gbj package:

- GbjExternalizable
- GbjObjectInputStream
- GbjObjectOutputStream
- GbxDelay
- GbxInput
- GbxOutput
- GbxSemaphore
- GbxTransport
- GbxUtl
- UnicodeFormatter

### GbjObject

GbjObject now extends GbjGciObject, and much of its core functionality has been moved to that class. In addition, design changes to support the GCI have required a number of API changes.

### Object Change Notification

The superclass GbjGciObject now provides support via the Java Observable/Observer API to notify customer application when the associated GS server object has changed, either due to GS Smalltalk code execution or committed transactions from other sessions. Refer to the GbjGciObject class documentation for details.

## Changes to Fields

A number of GbjObject fields have been moved to GbjGciObject/Report and/or are now protected (Customer applications can no longer directly access them and must use accessor methods). These include:

- ▶ `oop` - Moved to GbjGciReport and now protected. Use `getOop()` to access.
- ▶ `session` - Moved to GbjGciObject and now protected. Use `getSession()` to access.
- ▶ `cachedValue` - Replaced in GbjGciReport with several new protected fields. The method `cachedValue()` has been added to simulate its behavior, although it should be considered deprecated.
- ▶ `logger` - Now a static member.

## Changes to Methods

The biggest change in GbjObject has been in the constructor API. New constructors have been added to allow creating GS objects directly from a number of java base classes. These include:

```
GbjObject(boolean, GbjSession)
GbjObject(char, GbjSession)
GbjObject(byte, GbjSession)
GbjObject(short, GbjSession)
GbjObject(int, GbjSession)
GbjObject(long, GbjSession)
GbjObject(float, GbjSession)
GbjObject(double, GbjSession)
GbjObject(String, GbjSession)
GbjObject(Date, GbjSession)
GbjObject(Calendar, GbjSession)
```

The need to provide object canonicalization means that applications can no longer freely create GbjObjects with any OOP. This, in addition to the constructor API change to allow direct creation of GS objects from java base classes, causes an undetected conflict with existing GBJ applications, where the original constructor `GbjObject(long, GbjSession)`, which was originally used to create a GbjObject with a particular OOP, now generates a GS object representing that long as a GS SmallInteger object.

**Application developers MUST review their application code and replace these `GbjObject()` constructor calls with the method call `GbjObject.withOop(long, GbjSession)`.**

## Obsolete Methods

```
public void save()
public void release()
public void setOop()
public void setOopFromLong()
protected void privSetValues()
protected final GbjObject cloneIn()
public long getOopAsLong()
```

## Deprecated Methods

```
public static GbjObject fromInteger() - use "new GbjObject(int,
GbjSession)"
public static GbjObject fromLong() - use "new GbjObject(long,
GbjSession)"
```

## Moved Methods

These methods are still available but are now inherited from GbjGciObject/Report.

```
public boolean equals(Object arg)
public int hashCode ()
public int intValue()
public int byteValue()
public long longValue()
public float floatValue()
public double doubleValue()
public boolean booleanValue()
public char charValue()
public String stringValue()
public String toString()
public boolean isSpecial()
public Date dateValue()
public long getOop()
```

## GbjSession

GbjSession now extends GbjGciSession, and much of its core functionality has been moved to that class. In addition, design changes to support the GCI here required a number of API changes.

### Changes to Fields

A number of fields have become obsolete due to the change from the gem broker-based object marshalling scheme to the GCI-based architecture, and have been removed.

```
protected final Vector clientMsgThreads
protected GbjObject connection
protected GbxInput input
protected GbxOutput output
protected GbxTransport transport
protected GbjObject iteratorClass
private Vector exportedClientObjects
```

### Obsolete Methods

```
boolean exportedReferenceExists()
Vector exportedReferences()
int exportedReferencesSize()
void releaseAllObjs()
void releaseObj()
void removeAllExportedReferences()
void removeExportedReference()
```

```
GbjCollection savedObjs()
protected void checkForError()
protected void checkForOutOfSequence()
protected void clientMsgReceived()
protected void eventReceived()
protected void forcablyCloseConnection()
protected GbjObject newObject();
long[] clientStatistics()
public String transactionMode() - use getTransactionMode()
public void setTrace()
public void connect(boolean) - use connect()
public int sessionId() - use getSessionId()
public boolean connected() - use isConnected()
```

### **Modified Methods**

public String getName() - the format of the returned default string has changed.

### **Deprecated Methods**

public GbjObject putInServer()

## **GbjParameters**

The following fields have been deprecated and no longer have any effect on the session:

```
String brokerMachine
int brokerPort
int connectTimeout
int timeout
```

## **GbjKernelObjects**

This class has been extensively changed for version 3.0. It now only contains stubs for a few of the most frequently accessed GemStone server objects. For all others, applications should replace:

```
session.kernel.SomeKernelClass
```

With the sequence:

```
session.doit("SomeKernelClass")
```

The returned GbjObject can be cached in the application to avoid the cost of retrieving it.

## **GbjIterator**

Constructor GbjIterator(GbjObject[] initialCache) removed.

Constructor GbjIterator(GbjObject onCollection, long startingIdx) added.

## GbxLogger

No longer using `com.gemstone.org.apache.log4j` logger. Replaced with `java.util.logging.Logger`.

## Deprecated Classes

The class `DoubleByteString` is no longer needed, due to changes in the design of GBJ; Java strings are now automatically converted to GemStone Strings or `DoubleByteStrings` as necessary. `DoubleByteString` remains in the package to minimize customer impact.

## Package `com.gemstone.tools`

This package is largely unchanged in the GBJ 3.0 release. The only user-noticeable change is in the `GbjLoginDialog` window, where the following obsolete `GbjParameter` fields have been removed:

- ▶ GemBroker Machine
- ▶ GemBroker Port
- ▶ Timeout
- ▶ Connect Timeout